AFRL-RW-EG-TP-2007-7419

SINGLE SENSOR SCHEDULING FOR MULTI-SITE SURVEILLANCE (PREPRINT)

David E. Jeffcoat (AFRL/RWGN)
Navigation and Control Branch
Air Force Research Laboratory
Munitions Directorate
Eglin AFB, FL 32542

Mesut Yavuz
Harry F. Byrd, Jr. School of Business
Shenandoah University
Winchester, VA 22601

NOVEMBER 2007

JOURNAL ARTICLE (PREPRINT)

AIR FORCE RESEARCH LABORATORY, MUNITIONS DIRECTORATE

- **Air Force Material Command** ■ **United States Air Force** ■ **Eglin Air Force Base**

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
XX-11-2007	JOURNAL ARTICLE (PREPRINT)	

4. TITLE AND SUBTITLE

SINGLE SENSOR SCHEDULING FOR MULTI-SITE SURVEILLANCE (PREPRINT)

5a. CONTRACT NUMBER N/A

5b. GRANT NUMBER N/A

5c. PROGRAM ELEMENT NUMBER 61102F

6. AUTHOR(S)

1. Dr. David E. Jeffcoat
2. Dr. Mesut Yavuz

5d. PROJECT NUMBER 2304

5e. TASK NUMBER AM

5f. WORK UNIT NUMBER 07

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

1. Air Force Research Laboratory
   Munitions Directorate
   AFRL/RWGN
   Eglin AFB, FL 32542-6810
2. Harry F. Byrd, Jr. School of Business
   Shenandoah University
   Winchester, VA 22601

8. PERFORMING ORGANIZATION REPORT NUMBER

AFRL-RW-EG-TP-2007-7419

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Munitions Directorate
AFRL/RWGN
Eglin AFB, FL 32542-6810

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL-RW-EG

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

SAME AS BLOCK 8

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION A: Approved for public release; distribution unlimited.
Approval Confirmation #AAC/PA 10-22-07-616; dated 22 October 2007.

13. SUPPLEMENTARY NOTES

See 'cover page' for pertinent metadata information.

14. ABSTRACT

This paper is concerned with scheduling a single sensor to visit a number of sites with possibly time-variant dynamics. The paper motivates and presents a mathematical model for a sensor scheduling problem arising in the context of military operations research. The contributions made in the paper include the development and comparison of both deterministic and stochastic sensor scheduling methods. Results obtained in the paper show that the proposed heuristic methods can be used in real-time. A computational study is also provided. A deterministic greedy method produces the lowest cost solutions among the heuristics tested. The greedy methods produce the most predictable schedules, while the stochastic methods produce solutions that are less predictable in terms of site revisit times.

15. SUBJECT TERMS

Surveillance, sensor scheduling, time-variant site dynamics.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			DR. DAVID E. JEFFCOAT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	26	**19b. TELEPHONE NUMBER** *(Include area code)*

Single sensor scheduling for multi-site surveillance

Mesut Yavuz [*] David Jeffcoat [†]

Abstract

This paper is concerned with scheduling a single sensor to visit a number of sites with possibly time-variant dynamics. The paper motivates and presents a mathematical model for a sensor scheduling problem arising in the context of military operations research. The contributions made in the paper include the development and comparison of both deterministic and stochastic sensor scheduling methods. Results obtained in the paper show that the proposed heuristic methods can be used in real-time. A computational study is also provided. A deterministic greedy method produces the lowest cost solutions among the heuristics tested. The greedy methods produce the most predictable schedules, while the stochastic methods produce solutions that are less predictable in terms of site revisit times.

Key words: Surveillance, sensor scheduling, time-variant site dynamics.

[*] Harry F. Byrd, Jr. School of Business, Shenandoah University, Winchester, Virginia, U.S.A., Email: myavuz@su.edu

[†] Air Force Research Laboratory, Munitions Directorate, Eglin AFB, FL, U.S.A., Email: david.jeffcoat@eglin.af.mil

1 Introduction

Sensors are ubiquitous in modern society, both because of the increased need for surveillance and information for real-time decisions and because of the decreased cost of computation, communication, and data storage. Research and development of sensors involves interdisciplinary studies with findings published in scientific journals such as *IEEE Sensors Journal*, *International Journal of Distributed Sensor Networks*, *Ad Hoc & Sensor Wireless Networks*, and *International Journal of Remote Sensing*.

Sensors have found wide application in both industrial (see Mandroli et al. (2006) and the references therein) and defense contexts (see Rice and Willstatter (2000), Yost and Washburn (2000) and Miller (2007)). This paper is concerned with defense applications. Also, among various types of sensors such as acoustic, chemical, biological and visual sensors, we are interested in visual sensors which may or may not possess an on-board image processing capability. We are particularly interested in the use of satellites and video cameras in surveillance missions. We also note that surveillance is required in both day- and night-time. Therefore, it is natural to assume that video cameras used for surveillance possess imaging infrared (IIR) capabilities.

Satellites can be equipped with advanced cameras (Aplin et al. 1997), and have recently been used to observe earth for events such as forest fires and volcanic eruptions (Damiani et al. 2005). Generally observation tasks assigned to satellites are non-repetitive and optimal operation of a satellite is achieved by maximizing total benefit from the performed observations (Pemberton and Galiber 2001, Vasquez and Hao 2001, Lemaitre et al. 2002). To make an observation a satellite has to focus on a site of interest. Therefore, between two consecutive observations, the satellite will require some time to focus on the new site.

Surveillance using video cameras has improved in recent years (see Hampapur et al. (2003) and Bagdanov et al. (2005) for examples). A single camera or a network of cameras

can be used to simply monitor a spot or track a moving object in a region of interest. The shape of a region, and the number and sizes of obstacles in the region can prohibit effective surveillance of the region using only one camera. In that case a network of cameras is needed. A distributed approach to manage the network is to equip each camera with an on-board image processing capability. A centralized approach is to use only one image processor which can be a human expert or an automated image processing algorithm.

In all the above-mentioned papers, the goal is to perform non-recurring tasks such as taking a snapshot of a region or tracking a moving object. This paper is concerned with a single image processor used to monitor multiple sites of interest in a region, via periodic observations. For convenience, in the rest of the paper, we refer to the sensing and image processing system as the *sensor* and the location of interest as the *site*. Although the sensor may stay at a single physical location throughout a surveillance mission, processing images from geographically separated sites one at a time can be considered equivalent to visiting those sites in some order. Hence, we refer to processing an image from a site as a "visit" by our "mobile sensor" to that site. We note that many real-life scenarios can be described to motivate the problem addressed in this paper; we present only two due to space considerations.

A geo-synchronous satellite can be used to monitor intrusions along a border. Sites of interest can be unprotected points or paths along the border that can be used by intruders. Obviously, dedicating a fixed sensor (a dedicated satellite) to each site would be very costly, whereas a single sensor can be used to efficiently monitor the entire region by observing the sites in a sequence. Another possible application of this type is to monitor uninhabited areas such as sea lanes, woods or abandoned facilities for suspicious activities.

In our second example we consider improvised explosive devices (IEDs) and quote a recent study by White (2007):

> In the first six months of 2006, there were more than 11,000 IED attacks in Iraq, compared to 5,607 in 2004 and 10,953 in 2005, according to the Iraq Coalition Casualty Count. The US-based Homeland Security Research's 'Global Counter-IED Markets and Technologies Forecast 2008 to 2012' reported that insurgents plant IEDs under roads and gravel in order to avoid bomb sensors, requiring field commanders to use a variety of countermeasures, including increased patrols and closed-circuit television.

The streets of a city connect residential and business neighborhoods and may have varying traffic flows. Some locations can be critical targets where camera towers might be placed to monitor the city. Processing images from one camera at a time, instead of equipping each camera with an image processing capability, can be a practical and efficient solution. In this scenario, image processing is performed to rapidly yield a "yes" or "no" answer to such questions as "is anyone entering or leaving a building?," "is there a car parked near a critical location?" or "is there human activity adjacent to a road?"

As discussed earlier, the image processor in these examples can be considered a mobile sensor. At this point, we also distinguish our work from that on fixed sensors. Challenges arising from the use of fixed sensors include determining which sensor types to use, the quantity and location of each sensor type, and how to communicate in a network of sensors. With a mobile sensor, on the other hand, our focus is on scheduling visits to the sites.

Tiwari et al. (2005) consider a single, mobile sensor and apply a result from control and communication theory (Sinopoli et al. 2004) to quantify when a single sensor is sufficient to maintain a reliable estimate of linearly evolving parameters in multiple areas of interest. Building on the feasibility criterion for a single sensor, Yerrick et al. (2007) develop stochastic sensor scheduling algorithms for this same problem. That is, they first obtain stationary probabilities of the sensor being at each site such that error estimates for all the sites remain

bounded. They then either use the stationary probabilities as the basis for a probabilistic sensor motion strategy or obtain a matrix of transition probabilities in a Markov-chain setting. Yavuz and Jeffcoat (2007) apply results from the deterministic scheduling literature to develop a deterministic sensor scheduling policy to minimize the cost of lost information. In all these papers, the activity rates of the sites are assumed time-invariant over some planning horizon. In this paper, we generalize the results of Yavuz and Jeffcoat (2007) to allow the rate of activity or the value of information from areas under surveillance to evolve as a function of time. We also design a computational experiment, propose both deterministic and probabilistic scheduling methods and present a computational comparison of the proposed methods.

The remainder of this paper is organized as follows. In Section 2 we introduce a mathematical model and construct a computational study. In Section 3 we propose both deterministic and stochastic sensor scheduling methods, as well as a hybrid of the two approaches. We also discuss exact optimization approaches and lower bound policies for the problem. Section 4 presents the results from our computational study, and, finally, Section 5 provides a summary and conclusions.

2 Problem definition

In this paper, we assume that the sensor requires a constant amount of time to visit a site and make an observation. For a geo-synchronous satellite this implies a constant effort to re-focus on a new site. For the image processing function, it is valid to assume a constant image resolution, thereby a constant time to process the image. For video cameras, the central image processor can instantly select the next camera (equivalently, the next image to process). Again, a common image resolution implies a constant image processing time, thereby justifying the assumption. Therefore, we build our mathematical formulation in

discrete-time where the time required to visit a site and process the image is taken as the "time-unit."

Let $x_{i,t}$ be the binary decision variable denoting whether the sensor is scheduled to visit site i at time t, and $y_{i,t}$ denote the last time site i was visited as of the end of time t. Note that $y_{i,t} = t$ only in time intervals in which the sensor visits site i. When the sensor is focused on site i, it updates the status of the site. In other words, we have perfect information about the site in that time step. Since the sensor cannot focus on more than one site at the same time, focusing on one site means losing information about the current states of the other sites. The extent of the information loss depends on the activity rate of a site. We can afford to ignore less active sites for a large number of time steps, whereas more active sites must be visited frequently. In this paper, we assume that there is no cost for movement or observation; our whole concern is the information loss.

In the time-invariant version of the problem we associate with each site i a fixed penalty a_i and a variable penalty b_i of information loss. More specifically, a fixed penalty of not visiting a certain site is incurred for each time step in which the sensor is away from the site. In addition, a variable penalty is incurred for each time unit that has passed since the sensor's last visit to that site, providing ever-increasing motivation for the sensor to return to a neglected site. Determining the values of the parameters is straightforward for many applications. Recall the city surveillance example. The variable penalty b_i of a site i might be the average number of people or vehicles passing through the site in a time-unit. Multiplying this by the number of time units since our last visit to this site yields the expected number of people or vehicles that have passed through the site while we were not watching. As it is reasonable to set the probability of having an incident proportional to the traffic count, this count can be appropriately scaled to estimate the probability of missing an incident while not watching the site. The fixed penalty a_i can then be used to incorporate long-term historical data into the model. If a site i has been involved in many incidents, a large fixed penalty is

associated with it, thereby encouraging the sensor to visit that site more frequently even if the rate of activity at the site is low.

We introduce time-variant site dynamics to the model by allowing $b_{i,t}$ to vary throughout the planning horizon. An increase in the variable penalty parameter (i.e., $b_{i,t} - b_{i,t-1} > 0$ for some t) means that site i is becoming more active and, hence, requires more attention from the sensor. In the city surveillance example, it is natural to observe increased activity rates in rush hours. Similarly, a decrease (i.e., $b_{i,t} - b_{i,t-1} < 0$ for some t) means that the activity level of site i is diminishing, so that the sensor may visit it less frequently. Since the fixed penalty parameter a_i represents long-term history, we do not make a_i time-variant in our study.

The objective function in our model minimizes the maximum penalty incurred for a sensor schedule defined over a finite time horizon. This is achieved by defining $C = \max_{i,t}\{a_i + b_{i,t}(t - y_{i,t})\}$. Minimizing the maximum, known as the *min-max* approach, allows a fair allocation of the sensor's attention to sites. Based upon a known planning horizon length

(T), we formulate the following integer linear program.

$$
\begin{aligned}
&\text{Minimize} \quad C && && (1)\\
&\text{Subject to} && && \\
&C + a_i x_{i,t} + b_{i,t} y_{i,t} \geq a_i + b_{i,t} t, && \forall i = 1,..,n; \forall t = 1,..,T && (2)\\
&\sum_{i=1}^{n} x_{i,t} = 1, && \forall t = 1,..,T && (3)\\
&y_{i,t} - y_{i,t-1} \leq t x_{i,t}, && \forall i = 1,..,n; \forall t = 1,..,T && (4)\\
&y_{i,t} \leq t, && \forall i = 1,..,n; \forall t = 1,..,T && (5)\\
&y_{i,0} = 0, && \forall i = 1,..,n && (6)\\
&C > 0, && && (7)\\
&x_{i,t} \in \{0,1\}, && \forall i = 1,..,n; \forall t = 1,..,T && (8)\\
&y_{i,t} \in \{0\} \cup \mathbb{Z}^{+}, && \forall i = 1,..,n; \forall t = 1,..,T && (9)
\end{aligned}
$$

The model is built as a fully linear model, that is, the objective function and constraints are all linear functions of the decision variables. Note that defining C as a variable and including it in a constraint is critical for the linearity of the formulation. The objective function of the model (1) simply aims to minimize the maximum penalty defined by the first constraint (2). More specifically, $C \geq a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t})$, for all i and t. Constraint (3) assures that the sensor visits exactly one site in each stage. Constraints (4) and (5) together assure that $y_{i,t}$ is updated only when the sensor is at site i, and remains constant at other times. Constraint (6) initializes variable y. Finally, constraints (7-9) define the decision variables C, x and y as nonnegative, binary and nonnegative-integer, respectively. Also note that, if all the input parameters a_i and $b_{i,t}$ are integers, the optimal objective function value (C^*) will be integer, as well.

In this paper, we construct five test instances with which we can test and compare our solution approaches. Each instance consists of $n = 5$ sites and a planning horizon of $T = 500$ time units. The first instance is time-invariant. The purpose of including this instance is to compare the performance of deterministic and stochastic approaches, a comparison that has not been addressed in the existing literature on the time-invariant version of the problem. The second and third instances represent monotonically changing site-dynamics. The rate of change in site-dynamics are slow and fast in these two instances, respectively. In both instances, the first two sites become more active with time, whereas the last two become less active. The last two instances represent a cyclic pattern in the site dynamics. In the former, site dynamics are positively correlated. That is, all sites become more or less active at the same time. In the latter, the first three sites and the last two sites constitute two groups and as the sites in the first group become more active, those in the second group become less active, and vice versa. The structures of the instances are presented in Tables 1 and 2. The changes of the monotonically changing instances (in Table 1) are given with a two-field notation (;) where the numbers before the ";" are the periods of changes and the number after the ";" is the net common change in those periods. The cyclic changes (in Table 2) are represented with a more compact notation. The 12 numbers within the brackets are the net changes in periods that are multiples of ten. More specifically, the first change takes place at time 10, next one at time 20, and so on. The first cycle is completed at time 120. We repeat the cycle four times, thus all changes are completed at time 480.

[Insert Table 1 about here.]

[Insert Table 2 about here.]

3 Methods

3.1 Deterministic approach

The deterministic approach we adopt in this paper is the well-known greedy method. It starts with a null solution, i.e., an empty sequence. It evaluates all n sites that can be visited in the first stage. The selection of the next site to visit is made to minimize the penalty of information loss, i.e., penalty of not visiting a site. In the later stages the method evaluates $n-1$ sites (all but the one just visited) for its next visit. These simple selection and update operations are repeated until a complete solution (a sequence of T stages) is obtained. The time complexity of this procedure is $O(nT)$. It can easily be shown that the method does not necessarily construct optimal sequences, hence is a heuristic. We refer to this simple greedy procedure as G.

We improve the performance of the greedy method by adding a look-ahead feature. The heuristic still evaluates $n-1$ possible sites to visit in each stage, but makes the decision based on the penalty observed in the next ℓ stages. The computational complexity of the enhanced procedure is $O(n^2\ell T)$. Larger ℓ values are expected to yield better (lower penalty) solutions on the average, although this is not guaranteed. On the other hand, the number of operations to perform increases linearly with ℓ, thereby rendering small ℓ values more computationally efficient. In our preliminary experiments we found $\ell = n$ works best for most problems. We call this procedure *greedy with look-ahead* and refer to it as GLA.

3.2 Stochastic approach

In this section we develop a general stochastic sensor motion framework. The framework is built on the idea of optimizing steady state probabilities of the sensor to be located at each site. We also define optimal visit periods for the sites which are tied to the steady state

probabilities.

Let $r_i \in R$ be the visit period to site i. For the sake of simplicity, we assume r_i does not have to be an integer and any $r_i > 0$ can be achieved in an ideal schedule. Then, the penalty of information loss at site i is $a_i + (r_i - 1)b_{i,t}$ at time t. Here, let us postpone the discussion of the time-variability of $b_{i,t}$ and reduce the parameter to b_i. In other words, let us consider a sufficiently small planning horizon with time-invariant site dynamics. Visiting site i every $r_i > 0$ periods is equivalent to spending $\pi_i = 1/r_i$ of the sensor's time at that site. Obviously, the more time spent at a site, the less penalty incurred at that site. Therefore, we can conclude that in an optimal schedule $\sum_i \pi_i = 1$; i.e., all the available time is utilized. We also know that in reality the sensor never stays at any site for two consecutive time steps. Thus, $r_i \geq 2$ (equivalently, $\pi_i \leq 0.5$) should be satisfied for each site i. Putting all together, we formulate the following non-linear program to obtain optimal stationary probabilities.

$$\text{Minimize} \qquad C = \max_i \left\{ a_i + \left(\frac{1}{\pi_i} - 1 \right) b_i \right\} \tag{10}$$

Subject to

$$\sum_{i=1}^{n} \pi_i = 1, \tag{11}$$

$$\pi_i \leq 0.5, \qquad \forall i = 1, .., n \tag{12}$$

$$C > 0, \tag{13}$$

$$\pi_i \in \mathbb{R}, \qquad \forall i = 1, .., n \tag{14}$$

This non-linear program is defined on continuous variables only and is easily solved. Utilizing constraint (12), we define a lower bound on the objective function value with $C^L = \max_i\{a_i + b_i\}$. Then, we set $C = C^L$ and calculate $r_i = (C - a_i)/b_i + 1$ and $\pi_i = 1/r_i$ for all i. Note that $\pi_i = 0.5$ for the sites with $a_i + b_i = C$ and $\pi_i < 0.5$ for the rest. If

$\sum_i \pi_i = 1$, then we have found the optimal solution, and, hence can terminate. If $\sum_i \pi_i < 1$, we again have found the optimal objective function value. In order to satisfy constraint (11), we arbitrarily select a site with $\pi_i < 0.5$ and increase its stationary probability by the amount needed to satisfy the constraint. Note that, in this case, in the initial solution we have exactly one site with $\pi_i = 0.5$, thus we can find a feasible solution after only one step. If $\sum_i \pi_i > 1$, on the other hand, the current C value is not feasible. In this case, any systematic search procedure (such as bisection search) can be used to find a $C > C^L$ value that makes $\sum_i \pi_i$ sufficiently close to 1.

Example 1. Consider the first test instance in Table 1 with (a_i, b_i) pairs (125,25), (155,15), (170, 10), (165, 5) and (95, 15) for the five sites, respectively. $C^L = \max\{150, 170, 180, 170, 110\} = 180$, and it yields $\sum_i \pi_i = 0.313 + 0.375 + 0.500 + 0.250 + 0.150 = 1.588$, in which case we have to search for a $C > C^L$. A candidate is $C = C^L \times 1.588 = 285.75$, yielding $\sum_i \pi_i = 0.135 + 0.103 + 0.080 + 0.040 + 0.073 = 0.430$. Consequently, we have to search for a C such that $180 < C < 285.75$. $C = 200$ yields exactly $\sum_i \pi_i = 1$ for this instance.

The above approach is developed for time-invariant site dynamics. Therefore, when the site dynamics change, the transition probabilities have to be re-calculated to reflect the changes.

Example 2. Consider the second test instance in Table 1 with (a_i, b_i) pairs equal to (125,25) for all five sites, and $C^L = 150$ initially. The optimal solution can be easily calculated as $C = 225$ and $\pi_i = 0.200$ for all sites. At time 20, the variable penalty of the first site increases to 30 units thereby calling for re-calculating the optimal stationary probabilities. The solution to this new sub-problem is $C = 228.83$ which is achieved by setting $\pi_1 = 0.224$ and $\pi_i = 0.194$ for all other sites. As seen from this example, the site with the increased variable penalty receives more sensor attention at the expense of the other sites.

Up to this point, we have discussed a simple stochastic sensor motion policy based on optimal stationary probabilities of the sensor being at a site. This simple policy is static. That is, the transition probabilities are constant over time. However, since the objective function is the minimization of maximum penalty, a site becomes more critical as the number of time steps away from it increases. Therefore, reducing the probability of visiting a site that has been visited recently and increasing the probability of visiting a site that is overdue can improve the quality of the solutions obtained. We elaborate on this point in the following discussion.

Recall that $y_{i,t}$ is an integer decision variable representing the last time site i was visited by the end of time t. When constructing a visit sequence one starts with $y_{i,0} = 0$ for all i. Furthermore, in the first stage ($t = 1$), the site to visit is not known, hence, one has to use $y_{i,1} = y_{i,0} = 0$ for all i. After selecting the site to visit in the first stage (say, i_1^*) the last visit time for that site can be updated ($y_{i_1^*,1} = 1$). Generalizing this to any stage t, we state that the decision is made using $y_{i,t} = y_{i,t-1}$ for all i, and after the site to visit is selected (i_t^*) the last visit time for that site is updated ($y_{i_t^*,t} = t$). Now we focus on the process of tuning the visit probabilities for each site.

Let q_i be a preference value associated with visiting site i in a given stage t. (We use q_i instead of $q_{i,t}$ for convenience.) As discussed earlier, in an optimal schedule the sensor never stays at the same site in two consecutive time steps. Accordingly, we set $q_{i_{t-1}^*} = 0$. For all other sites we calculate the visit preference using $q_i = \pi_i \times \left(\frac{t-y_{i,t}}{r_i}\right)^k$. Here, k is a user-defined parameter representing the weight of the adjustment factor $\left(\frac{t-y_{i,t}}{r_i}\right)$, which is smaller than 1 for the sites that have been visited within their ideal (and expected) visit periods and greater than 1 for the overdue sites. Using a larger k value puts more emphasis on visiting the overdue sites. Also note that using $k = 0$ disables the adjustment factor, or, equivalently, uses the stationary probabilities in a static manner. When the preference values are calculated for all sites, they can be converted to probabilities with $p_i = q_i/Q$ for

all i where $Q = \sum_{i=1}^{n} q_i$. The pseudo-code of the stochastic approach is presented in Figure 1.

[Insert Figure 1 about here.]

3.3 A hybrid approach

The stochastic approach described in the previous section is based on optimized stationary probabilities of the sensor being focused on each site. Therefore, after the optimization of the stationary probabilities, the method pays almost no attention to the objective function contribution of the decisions made in each stage of the sequence. The greedy method described earlier, on the other hand, makes the decisions solely based on the objective function contribution. In this section, we present a hybrid of the two approaches which can also be called *probabilistic greedy method*.

In the hybrid method, one first calculates the penalty of not visiting each site $c_i = a_i + b_{i,t}(t - y_{i,t})$. The next step is to calculate preference values to visit each site with $q_i = \left(\frac{c_i}{c_{\max}}\right)^k$. Here, $c_{\max} = \max_i\{c_i\}$ is the maximum penalty of not visiting a site. Also, the implementation is similar to the stochastic method as $q_{i^*_{t-1}} = 0$ is set for the site just visited, and $Q = \sum_i q_i$ is used to transform the visit preferences to probabilities ($p_i = q_i/Q$ for all i). The pseudo-code of the hybrid approach is presented in Figure 2.

[Insert Figure 2 about here.]

3.4 Exact optimization and lower bounds

Yavuz and Jeffcoat (2007) show that the time-invariant version of the problem is NP-Hard. Since allowing time-variant site dynamics is a generalization of the time-invariant problem, the time-variant problem studied in this paper is NP-Hard, as well. Therefore, exact optimization approaches are likely to be prohibitively time-consuming. To test this hypothesis

we used a commercially available optimization software, C-Plex 9.1, and observed that none of the five test instances (in Tables 1 and 2) were solved within 30 minutes. Therefore, we focus on obtaining lower bounds for the objective function values of the five test instances addressed in this paper.

As stated earlier, the sensor has to spend at least one time unit away from a site i, thus incurring a cost of $a_i + b_{i,t}$ at that site. Therefore, $C^L = \max_{i,t}\{a_i + b_{i,t}\}$ is a valid lower bound for the problem.

It is easy to see that in many real-life instances (especially with a large number of sites) the sensor has to spend more than one time-step away from a site. Therefore, C^L is not necessarily a tight lower bound. To obtain a better lower bound, we focus on sub-problems and then solve each sub-problem with C-Plex 9.1. In our experiments, we found that sub-problems with $T = 16$ stages were easily solvable. Thus, we formulated 50 sub-problems for each instance such that sub-problem τ covers the stages $[(\tau-1)\times 10+1, .., \min\{(\tau-1)\times 10+16, 500\}]$, $\tau = 1, .., 50$. Note that this approach does not provide a perfect decomposition, since the sub-problems are overlapping. The objective function values of the sub-problems are compared and their maximum is used as a lower bound to the problem. Since the objective is of the min-max type, the overlap between the subproblems does not inversely impact the lower bound, but may in fact strengthen it.

The first test-instance is time-invariant, thus all the sub-problems are identical. In fact, the optimal solution to this instance can be easily constructed. If integer periods r_i to visit sites $i = 1, .., 5$ exist such that $a_i + (r_i - 1)b_i = C$ for all i and r_i are achievable in a cyclic schedule, then C is the optimal objective function value. The proof is based on the observation that if one site is given more time, then the other sites must receive less time and their penalties must be larger. Consequently, the maximum of the penalties is minimized by a solution that makes all the penalties equal. $C = 40$ is achievable for the first instance by setting $r_1 = r_2 = r_3 = 4$ and $r_4 = r_5 = 8$, and repeating the visit sequence

$1-2-3-4-1-2-3-5$.

4 Results

In this section we report and discuss the results from our computational study. All five-instances are first divided into sub-problems and lower bounds are obtained. Then, both the simple greedy (G) and the greedy with look-ahead (GLA) methods are implemented. The stochastic method is implemented with four different levels of the input parameter $k = 0, 1, 2, 3$. Recall that $k = 0$ is equivalent to using the optimized stationary probabilities in a static fashion. Finally, the hybrid method is implemented with three different levels of the input parameter $k = 1, 2, 3$. All together, we experiment with ten methods: LB, G, GLA, S(0), S(1), S(2), S(3), H(1), H(2) and H(3).

The stochastic and hybrid methods construct solutions in a probabilistic fashion, thus we run them 100 times on each instance and report average-case results.

All the methods take negligible time on a desktop computer with a P4-3.4GHz CPU and 2GB of memory. Therefore, we do not report on the computation time taken by the methods. We compare the methods using two criteria, namely solution quality and solution variability.

Let C^{LB} be the lower bound value calculated for the objective function. Also, let C^X be the objective function value of the solution obtained by some method X. Then, the solution quality for method X is calculated as $\frac{C^X - C^{LB}}{C^{LB}} \times 100\%$, so that smaller values represent better solutions.

Our variability criterion is adapted from Corominas et al. (2007). We analyze a solution, i.e., a complete sequence of 500 stages, and note the distances between two consecutive visits to each site. These distances are stored in variables $\eta_{i,\upsilon}$, $i = 1, .., n$ and $\upsilon = 1, .., \Upsilon_i$, where Υ_i is the number of visits to site i. When the visit intervals are determined, their average

($\bar{\eta_i}$) is calculated. Finally, the (per stage) variability of the visit intervals is calculated with $\sum_i \sum_v (\eta_{i,v} - \bar{\eta_i})^2/T$. Again, for the stochastic and hybrid methods, we report the averages of the 100 runs performed with each setting of the method. The results are presented in Table 3.

[Insert Table 3 about here.]

The results presented in Table 3 reveal that relative deviation from the lower bound and variability are positively correlated. Stated another way, high variability tends to be associated with poor solution quality.

One quick observation from the table is the extremely high variability and poor solution quality of S(0) compared with the other methods. This is due to its static nature in using the stationary probabilities to construct a sequence.

At the opposite extreme are the deterministic methods G and GLA. They both yield low variability and good solution quality. When we compare the two with each other, we observe that GLA consistently yields a better solution quality and a greater variability.

For the probabilistic methods, the first result we observe is that as k increases both methods behave much like the greedy method. In other words, smaller k values represent a higher degree of randomness and larger k values represent an almost deterministic selection of the next site to visit. When we compare the stochastic and the hybrid methods with each other, we observe that S(k) yields a better solution quality and a smaller variability than H(k), for $k = 1, 2, 3$.

In many real-life scenarios a high variability may be desired, since it is equivalent to unpredictability of the schedule. In such scenarios, we infer two results. First, GLA dominates G with its better solution quality and greater variability. Second, GLA and the probabilistic methods are all appropriate and one should select a method based on the degree of variability and solution quality desired. On the other hand, in some applications, e.g., search and

rescue missions, a low variability is desired. Similar scenarios arise in the fields of manufacturing and supply chain management (see Corominas et al. 2007). In such scenarios, the results obtained in this paper show that deterministic methods are preferable to probabilistic methods.

5 Conclusions

In this paper we have addressed the problem of scheduling a single sensor to visit a number of sites with possibly time-variant dynamics. This problem is an extension of the sensor scheduling problem that has been studied previously with only time-invariant site dynamics. The contributions made in the paper include the development of both deterministic and stochastic sensor scheduling methods and a comparison of the two approaches.

For testing purposes we created five test instances each with different characteristics. The results show that the methods proposed in this paper are all time-effective. When compared, the methods show a trade-off between solution quality and variability of the schedules obtained. Therefore, the selection of the most appropriate method should be made by the user based on the desired solution quality and variability.

References

P. Aplin, P.M. Atkinson, and P.J. Curran. Fine spatial resolution satellite sensors for the next decade. *International Journal of Remote Sensing*, 18(18):3873–3881, 1997.

A.D. Bagdanov, A. del Bimbo, and F. Pernici. Acquisition of high-resolution images through online saccade sequence planning. In *VSSN '05: Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 121–130, New York, NY, USA, 2005. ACM Press. ISBN 1-59593-242-9. doi: http://doi.acm.org/10.1145/1099396.1099419.

A. Corominas, W. Kubiak, and N.M. Palli. Response time variability. *Journal of Scheduling*, 10: 97–110, 2007.

S. Damiani, G. Verfaillie, and M.-S. Charmeau. A continuous anytime planning module for an autonomous earth watching satellite. In *Proceedings of the ICAPS-05 Workshop on Planning under Uncertainty for Autonomous Systems*, Monterey, CA, USA, 2005.

A. Hampapur, L. Brown, J. Connell, S. Pankanti, A. Senior, and Y. Tian. Smart surveillance: Applications, technologies and implications. In *Proceedings of the 2003 Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing, and the Fourth Pacific Rim Conference on Multimedia*, volume 2, pages 1133–1138, Hawthorne, NY, USA, Dec. 2003.

M. Lemaitre, G. Verfaille, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6:367–381, 2002.

S.S. Mandroli, A.K. Shrivastava, and Y. Ding. A survey of inspection strategy and sensor distribution studies in discrete-part manufacturing processes. *IIE Transactions*, 38:309–328, 2006.

J.G. Miller. A new sensor allocation algorithm for the space surveillance network. *Military Operations Research*, 12(1):57–70, 2007.

J.C. Pemberton and F. Galiber. A constraint-based approach to satellite scheduling. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 57:101–114, 2001.

R.E. Rice and K. Willstatter. On determining an optimum C4ISR architecture: the sensor-platform allocation analysis tool (SPAAT). *Military Operations Research*, 5(2):37–64, 2000.

B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, 2004.

A. Tiwari, M. Jun, D. Jeffcoat, and R. Murray. The dynamic sensor coverage problem. In *Proceedings of the 16th International Federation of Automatic Control (IFAC) World Congress*, Prague, Czech Republic, July 2005.

M. Vasquez and J.-K. Hao. A “logic-constrained” knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Computational Optimization and Applications*, 20:137–157, 2001.

A. White. Report spotlights expanding threat of IEDs in afghanistan and iraq. available on-line at http://www8.janes.com, posted 25 July 2007, 2007.

M. Yavuz and D. Jeffcoat. An analysis and solution of the sensor scheduling problem. In *Proceedings of the 7th International Conference on Cooperative Control and Optimization*, Gainesville, FL, 2007.

N. Yerrick, M. Yavuz, and D. Jeffcoat. Two sensor motion models for the dynamic sensor coverage problem. *Military Operations Research*, 12(2):55–64, 2007.

K.A. Yost and A.R. Washburn. Optimizing assignment of air-to-ground assets and BDA sensors. *Military Operations Research*, 5(2):77–91, 2000.

Table 1: Test instances 1-3

Instance	i	$a_{i,0}$	b_i	Changes in $a_{i,t}$
1	1	125	25	none
	2	155	15	none
	3	170	10	none
	4	165	5	none
	5	95	15	none
2	1	125	25	(20, 140, 180, 300, 340; +5)
	2	125	25	(60, 120, 200, 260, 400; +2)
	3	125	25	none
	4	125	25	(80, 160, 220, 320, 380; -2)
	5	125	25	(40, 100, 240, 280, 360; -5)
3	1	125	25	(60, 140, 220, 300, 380; +5)
	2	125	25	(25, 55, 85, 115, 145, 175, 205, 235, 265, 295, 325, 355; +2) (370; +1)
	3	125	25	none
	4	125	25	(30, 50, 70, 90, 110, 130, 150, 170, 190, 210, 230, 250; -2) (260; -1)
	5	125	25	(80, 160, 240, 320, 400; -5)

Table 2: Test instances 4-5

Instance	i	$a_{i,0}$	b_i	Cyclic changes in $a_{i,t}$
4	1	125	25	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
	2	100	20	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
	3	150	25	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
	4	175	15	(+6, +4, +2, -2, -4, -6, -6, -4, -2, +2, +4, +6)
	5	125	30	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
5	1	125	25	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
	2	100	20	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
	3	150	25	(+3, +2, +1, -1, -2, -3, -3, -2, -1, +1, +2, +3)
	4	175	15	(-6, -4, -2, +2, +4, +6, +6, +4, +2, -2, -4, -6)
	5	125	30	(-3, -2, -1, +1, +2, +3, +3, +2, +1, -1, -2, -3)

Figure 1: Pseudo-code for Algorithm `Stochastic`(k)

Algorithm `Stochastic`(k)
BEGIN
1. Set $y_{i,0} = 0, i = 1, 2, .., n$ and $i^*_0 = 0$.
2. FOR $t = 1$ to T, increase t by 1
    BEGIN
    3. IF site dynamics has changed THEN
        Obtain π_i and r_i for all i by solving the NLP in page 6.
    4. Set $Q = 0$.
    5. FOR $i = 1$ to n, increase i by 1
        BEGIN
        6. Set $y_{i,t} = y_{i,t-1}$.
        7. IF $i = i^*_{t-1}$
        THEN
            8. Set $q_i = 0$.
        ELSE
            9. Set $q_i = \pi_i \left(\frac{t - y_{i,t}}{r_i}\right)^k$.
        10. Update $Q \leftarrow Q + q_i$.
        END
    11. FOR $i = 1$ to n, increase i by 1
        BEGIN
        12. Set $p_i = \frac{q_i}{Q}$.
        END
    13. Select i^*_t w.r.t. probabilities $p_i, i = 1, .., n$.
    14. Set $y_{i^*_t,t} = t$.
    END
END.

Figure 2: Pseudo-code for Algorithm `Hybrid`(k)

Algorithm `Hybrid`(k)
BEGIN
1. Set $y_{i,0} = 0, i = 1, 2, .., n$ and $i_0^* = 0$.
2. FOR $t = 1$ to T, increase t by 1
    BEGIN
    3. Set $c_{\max} = 0$.
    4. FOR $i = 1$ to n, increase i by 1
        BEGIN
        5. Set $y_{i,t} = y_{i,t-1}$.
        6. IF $i = i_{t-1}^*$
        THEN
            7. Set $c_i = 0$.
        ELSE
            BEGIN
            8. Set $c_i = a_i + b_{i,t}(t - y_{i,t})$.
            9. IF $c_{\max} < c_i$ THEN
                10. Update $c_{\max} \leftarrow c_i$.
            END
        END
    9. Set $Q = 0$.
    11. FOR $i = 1$ to n, increase i by 1
        BEGIN
            12. Set $q_i = \left(\frac{c_i}{c_{\max}}\right)^k$.
        13. Update $Q \leftarrow Q + q_i$.
        END
    14. FOR $i = 1$ to n, increase i by 1
        BEGIN
        15. Set $p_i = \frac{q_i}{Q}$.
        END
    16. Select i_t^* w.r.t. probabilities $p_i, i = 1, .., n$.
    17. Set $y_{i_t^*,t} = t$.
    END
END.

Table 3: Summary of the results

		Method								
Instance	Measure	G	GLA	S(0)	S(1)	S(2)	S(3)	H(1)	H(2)	H(3)
1	Rel. Dev.	7.50 %	2.50 %	241.93 %	89.48 %	58.83 %	47.43 %	129.75 %	92.58 %	76.20 %
	Variability	0.01	0.01	22.21	5.80	3.33	2.18	8.48	6.67	5.46
2	Rel. Dev.	14.29 %	12.24 %	251.35 %	83.62 %	55.87 %	42.91 %	146.62 %	101.26 %	76.48 %
	Variability	2.41	3.09	20.19	6.42	4.44	3.49	8.06	7.18	7.38
3	Rel. Dev.	7.44 %	3.31 %	241.73 %	83.89 %	53.59 %	40.88 %	154.75 %	101.65 %	75.05 %
	Variability	3.08	5.37	18.72	6.48	4.98	4.14	8.83	9.47	11.38
4	Rel. Dev.	2.91 %	2.09 %	231.30 %	85.68 %	53.22 %	41.15 %	105.93 %	75.83 %	57.40 %
	Variability	0.58	1.02	20.65	5.60	3.30	2.28	7.49	5.47	4.13
5	Rel. Dev.	10.20 %	8.16 %	257.56 %	89.32 %	58.82 %	46.00 %	123.38 %	89.33 %	66.35 %
	Variability	2.34	5.21	23.54	7.42	5.12	4.01	7.54	5.65	4.41